Applicant: Stein et al.
For: SINGLE INSTRUCTION MULTIPLE DATA ARRAY CELL

FIELD OF THE INVENTION

5   This invention relates to a single instruction multiple data array cell for processing a direct

access memory data stream.

RELATED APPLICATIONS   /Serial No. 60/350,398

This application claims priority over U.S. Provisional application entitled SINGLE

10   INSTRUCTION MULTIPLE DATA (SIMD) ALU ARRAY to Kablotsky et al., filed January 21,

2002.

BACKGROUND OF THE INVENTION

The high performance required for real-time processing in communications and

15   multimedia applications stresses processor architectures in many different ways. The Single

Instruction Multiple Data (SIMD) parallel-processing model that exploits the application's

properties of natural parallelism is considered the most acceptable way to deliver the high

performance need for both today's and future applications. The SIMD model assumes a plurality

of processing cells. Each cell may include various combinations of hardware: address generators,

20   multipliers, arithmetic logic units (ALUs), memory, registers, and sequencers. Generally, as the

investment in hardware increases, the speed of processing goes up, but so does the power

requirement, cost, and die area required. Cost and area are always considerations in hand held

devices such as personal digital assistants (PDAs), cell phones, and other wireless handset

terminals. Many different approaches have been used to address these problems. One approach,

AD-303J   2
JSI:ci

APPLICATION

FOR

UNITED STATES LETTERS PATENT

10   Be it known that we, Yosef Stein, residing at 4 Turning Mill Road, Sharon,

Massachusetts 02067 and being a citizen of Israel and Joshua A. Kablotsky, residing at 2191

Bay Road, Sharon, MA 02067 and being a citizen of the United States, have invented a certain

new and useful

SINGLE INSTRUCTION MULTIPLE DATA ARRAY CELL

15   of which the following is a specification:

EL861890280US

Applicant: Stein et al.
For: SINGLE INSTRUCTION MULTIPLE DATA ARRAY CELL

<div align="center">

FIELD OF THE INVENTION

</div>

5   This invention relates to a single instruction multiple data array cell for processing a data

stream.

<div align="center">

RELATED APPLICATIONS

</div>

This application claims priority over U.S. Provisional application entitled SINGLE

10   INSTRUCTION MULTIPLE DATA (SIMD) ALU ARRAY to Kablotsky et al., filed January 21,

2002.

<div align="center">

BACKGROUND OF THE INVENTION

</div>

The high performance required for real-time processing in communications and

multimedia applications stresses processor architectures in many different ways. The Single

Instruction Multiple Data (SIMD) parallel-processing model that exploits the application's

properties of natural parallelism is considered the most acceptable way to deliver the high

performance need for both today's and future applications. The SIMD model assumes a plurality

of processing cells. Each cell may include various combinations of hardware: address generators,

20   multipliers, arithmetic logic units (ALUs), memory, registers, and sequencers. Generally, as the

investment in hardware increases, the speed of processing goes up, but so does the power

requirement, cost, and die area required. Cost and area are always considerations in hand held

devices such as personal digital assistants (PDAs), cell phones, and other wireless handset

terminals. Many different approaches have been used to address these problems. One approach,

known as the very long instruction word (VLIW), VLIW architecture increases the speed by

packing multiple operations into a single instruction word, which is then executed in parallel as a

very wide instruction unit. However, this requires a very large register capacity and memory to

store those instructions. The programming limitations of VLIW processors require engineers to

5   use very low assembly programming to achieve high performance. Such programming requires

specialized knowledge of the hardware architecture, which can be extremely complex. An

alternative architecture to VLIW is Single Instruction Multiple Data (SIMD). This model assumes

an array of processing cells each executing the same sequence of instruction on their local data.

The key advantages of this approach are a reduction in overall hardware complexity, design

10   regularity, the enhancement of computing resources and simplified path to software development.

These come from the fact that only a single instruction-decode-and-dispatch is required. An

example for such an array is the reconfigurable ALU array, offered by Elixent Limited of Bristol,

England, which uses an array of four bit ALUs and register/buffer blocks. The ALUs are

interwoven with adjacent cross bar switches. This results in a highly reconfigurable array but

15   requires a large die area, a long time to accomplish the reconfiguring and a large number of buses.

Yet another reconfigurable approach, the PACT, XPP, parallel processes the data by using

processing array elements (PAE) where each of the PAE's uses an ALU and a few registers but is

limited to only ALU types of operations without multiplication. In still another approach, using a

multiple instruction multiple data (MIMD) array, each cell includes a full digital signal processor

20   (DSP) that can change from a simple DSP up to a VLIW type; each DSP includes data address

registers (DAGs), a compute block comprising at least one ALU, a shifter and a 16-bit multiplier

with a register file, an instruction decoder and a sequencer. These systems are fast and versatile

but they require very high power and very large die area. See the reconfigurable ALU array (RAA)

AD-303J   3
JSI:ci

at www.elixent.com. See also the XPP architecture at www.PACTCORP.com.

BRIEF SUMMARY OF THE INVENTION

It is therefore an object of this invention to provide an improved single instruction multiple

5   data (SIMD) array cell for processing a data stream.

It is a further object of this invention to provide a SIMD array cell which uses less die area.

It is a further object of this invention to provide a SIMD array cell which eliminates the

need for data address generators (DAG's) and all the associated Base, Index, Length, and Modify

(B, I, L, M) registers.

10   It is a further object of this invention to provide a SIMD array cell which eliminates the

need for a multiplier circuit by using a shift and add/subtract circuit.

It is a further object of this invention to provide a SIMD array cell which eliminates the

need for registers in the arithmetic logic circuit by storing the data directly in memory.

It is a further object of this invention to provide a SIMD array cell which eliminates the

15   need for loop logic and the associated registers by operating with absolute addresses.

It is a further object of this invention to provide a SIMD array cell which uses less power

by replacing multiplication with additions/subtractions and very short shifts.

It is a further object of this invention to provide a SIMD array cell which powers down

cells not required for a particular operation.

20   It is a further object of this invention to provide a SIMD array cell which eliminates the

need for an in cell sequencer by using a single instruction-decode-and-dispatch mechanism.

The invention results from the realization that a truly improved single instruction multiple

data array for processing a data stream, which is faster, and uses less power and less die area, can

be achieved using a plurality of cells, which need no address generators or associated registers, each of which cells have a memory that stores a predetermined region of the data stream, a location register for representing the size and location of that region, a unique identification number and an arithmetic logic unit responsive in a load mode to the identification number and a

5   single command word common to all cells to compute the unique start position for its cell for receiving the predetermined region of the direct memory access data stream. In an execution mode the command word includes an address field applicable to all cells, a data field and an instruction to be performed by the arithmetic logic unit. The arithmetic logic unit in each cell performs the instruction directly on the local value at that address in its memory with the data in the data field.

10   This invention features a single instruction multiple data (SIMD) array cell for processing a data stream, the array including a plurality of cells, each cell including a memory circuit for storing a predetermined region of the data stream, a location register circuit for representing the size and location of the predetermined region of the data stream, a unique identification number stored in its ID register circuit and an arithmetic logic unit responsive to the identification number and a single

15   command common to all cells in a load mode to compute a unique start position for its cell for receiving the predetermined region of the data stream.

In a preferred embodiment, the arithmetic logic unit may include an adder circuit. The arithmetic logic unit may include a shifter circuit for performing multiplication and accumulation in combination with the adder circuit. The arithmetic logic unit may include an arithmetic logic

20   circuit. The command word may include in an execution mode an address field applicable to all cells. The arithmetic logic unit may operate directly on the values stored at that address in its cell memory with the data in the data field. The command word in the execution mode may include an instruction field for operating the arithmetic logic unit. The arithmetic logic unit may include a

AD-303J   5
JSI:ci

multiplexor for presenting input from the memory circuit, the command word, and the output of the arithmetic logic unit. Each cell may include a condition code register and the arithmetic logic unit may respond to the unique identification number and the condition code register to control the condition of the cell. The location register circuit may store a start position for the predetermined

5   region to be stored in its memory, the length of the direct memory access data stream to be stored, and at least one dimension of the data stream. The location register may store the vertical and horizontal dimensions of the data stream. The command word in the execution mode may establish the size and location of the predetermined region in the location register circuit. The command word may include an address field for addressing locations in the predetermined region

10   of interest in the memory circuit. The command word in the execution mode may include a data field for operating the arithmetic logic unit. The condition code register and arithmetic logic unit may respond to the unique identification number, the data field, to control the condition of the cell.

BRIEF DESCRIPTION OF THE DRAWINGS

15   Other objects, features and advantages will occur to those skilled in the art from the following description of a preferred embodiment and the accompanying drawings, in which:

Fig. 1 is a schematic diagram of an single instruction multiple data array superimposed on a portion of an image;

Fig. 2 is an exploded schematic diagram showing the memories associated with each of the

20   array cells in Fig. 1 arranged to depict the portion of the picture stored in them;

Fig. 3 is a schematic diagram of a single instruction multiple data array cell according to this invention;

Fig. 4 is a schematic diagram of the single instruction multiple data array employing a

AD-303J   6
JSI:ci

number of cells according to this invention;

Fig. 5 is a schematic illustration of the format of the command word used with this invention;

Fig. 6 is a view similar to Fig. 5 depicting variations of the command in the execute mode for loading the location registers of Fig. 3;

Fig. 7 is a view similar to Fig. 5 showing variations in the command in the load mode for access of a data stream;

Fig. 8 is a schematic diagram illustrating the derivation of certain vertical and horizontal dimensions and start position stored in the location register circuit;

Fig. 9 is a view similar to Fig. 5 of the command word in the execute mode for carrying out a single instruction simultaneously on the local data stored at the same address in each memory of each cell all operated on by the same global data dispatched in the command;

Figs. 10 and 11 are views similar to Fig. 5 showing the variations in the command during a condition code operation to put certain selected cells into sleep condition; and

Fig. 12 illustrates operation in a cell overlap arrangement.

PREFERRED EMBODIMENT

There is shown in Fig. 1 a single instruction multiple data array 10 capable of storing, in this example, one quarter of the total image 12. Array 10, in this embodiment, includes an 8 x 8 array of 64 cells 14, each of which contains a memory unit shown schematically at 16, Fig. 2, arranged in array 10 to represent the upper left quarter of image 12.

Each cell 14a, Fig. 3 includes, in addition to memory 16a, a location register circuit 18, an arithmetic logic unit 20, a condition code register 22, and an identification number register 24.

Memory 16a is a region of interest memory of 256 bytes or pixels. That is, it may hold an area

sixteen pixels on a side representing the region of interest (ROI) of image 12, Fig. 1. The location

and size of the region of interest stored in memory 16a is represented in location register circuits

18 which contains the starting point for the data, the length of the data stream, and at least one

5   dimension such as the vertical or horizontal or both. For two-dimensional data streams such as

images, both would be utilized; for signal processing where the data stream is one-dimensional,

only one, for example the horizontal value, need be available. Arithmetic logic unit 20 includes an

arithmetic logic circuit 26, an adder 28, and a shifter 30 so that both addition and multiplication by

additions/subtractions and shifting can be accomplished. A 4:1 multiplexor 32 provides one input

10   to arithmetic logic circuit 26. That input to arithmetic logic circuit 26 may be the fed-back signal

at its own output from adder 28 or the signal from the condition code register 22 or the data from

the ROI memory 16a or the identification number from identification number register 24. The

identification number register 24 holds a number which is unique for each cell, for example

referring again to Fig. 1, the top row of cells 14 may be identified as 0-7, the second row as 8-15,

15   the third row as 16-23 and so on till the last row of 56-63. All of the cells 14, Fig. 4, are connected

to their neighboring cells and are also serviced by a command/access bus 40. Bus 40 may be a

direct memory access bus or the data may come from a processor or core itself in either the load or

execution mode. The single instruction multiple data arrangement is delivered by 2:1 multiplexor

42 which provides, for example, either the direct memory access (DMA) data stream on line 44 in

20   the load mode or a command word on line 46 in the execute mode. Bus 40 is a 32-bit bus. The

format of the command 50, Fig. 5, includes a 16 bit data field 52 from bit 0 to bit 15, an 8 bit

address field 54 from bit 16 to bit 23, an instruction field 56 from bit 24 to bit 30, and a 2 bit code

condition field 58 from bit 30 to bit 31.

AD-303J   8
JSI:ci

In operation, in the execute mode, Fig. 6, in order to define the (ROI-DMA) parameters the command 50a will hold in the instruction field 56a either the instruction Ld Hlen, Ld Vert or Ld Horiz to variously load the Hlen, Vert, and Horiz values, as shown in Fig. 8, in location register circuit 18, Fig. 3. Because all array processing elements are processing the same patch size, the

5　particular length or size for those characteristics are simultaneously loaded in the data field 52a. The address field 54a and condition code field 58a are not used in this operation. Horiz is the patch horizontal length to be stored in each of a memory 16, Vert is the patch vertical length of each memory 16, which in this embodiment are sixteen because each memory 16 is composed of a square sixteen by sixteen byte matrix. Hlen is the combined length of the horizontal portion of the

10　data stream being deposited in all memories 16 or each array row. For example, in Fig. 8 where cells 0, 1, 8, and 9 are being loaded in a raster pattern the data must be received in one hundred twenty eight bytes/line, line after line until the memories in all four cells 0, 1, 8, 9 have been loaded.

In loading mode the tightest rectangle that contains all the data points (NxNxROI) needed

15　to be processed by the NxN array is raster broadcasted across the arrays DMA bus. Each one of the array members receives the broadcast data and grabs its own region of interest data according to the settings of his own location register circuit. In, Fig. 7, the command word 50b provides a sequence of instructions to program each cell's register starting point as a function of its unique identification number to a different value using the same dispatched instruction. The start value is

20　installed in location register circuit 18 using the command 50a, Fig. 6, where the instruction sequence establishes the start point of the region of interest in each memory 16, by combining the identification number from identification number register 24 with the data 52a from the command word 50a, Fig.6, in accordance with the instructions 56a.

AD-303J　　　　　　　　　　　　　　　9
JSI:ci

For example, in order to set the starting point of cells 0, 1, 8, 9, ... of Fig. 8, the following instruction sequence Strt = Alu = (Id>>3) << 9 + (ID && 7) << 4, (where shift left is indicated by <<, shift right by >> and bit-wise and by &&) should be executed by the arithmetic logic unit 26 shifter 30 and adder 28 in each cell. To illustrate for Id = 9 Start = 512+16, because, Strt = Alu = Fix(Id/8)*512+(Id && 7)*16 = 512+16. For the next row beginning with the identification number 16, using the same formula will get Strt = Alu = Fix(Id/8)*512+(Id && 7)*16 = 2*512+0*16, and so on.

In the execution mode, the instruction field 56c, Fig. 9, instructs the ALU to operate on the data in the memory at address field 54c with the data in data field 52c: ALU+ = MEM(address) << data is a single instruction which tells each cell to go to that address in its memory and to operate on it with the data in field 52c. The operation is typically a shifting and add to effect multiplication. The instruction field 56c can have up to 64 different types where the most common are MIN, MAX, ACS (Add Compare Select) and all the bit-wise operations like OR, AND, and XOR. It is the use of this single instruction on multiple data which dramatically increases the speed of operation. The common command word holds a single instruction, a single address and a single piece of data. The single address addresses the same place in each memory of each cell; the single instruction causes each ALU to operate in accordance with the single instruction on the data at that address in each memory with the piece of data in the data field of the common command word. The operation takes place simultaneously in all cells at the same time. In this embodiment, where all sixty-four cells are simultaneously processing the data they have at that specific memory address with the data supplied by command 50c, the time of operation is reduced by a factor equal to the number of cells involved: where sixty-four cells are operating simultaneously, the operation is sixty-four times faster.

Condition code register 22 maintains the status of the ALU in its particular cell. For example, when command 50d, Fig. 10, contains the instructions ALU=ID-data that instruction is held in the instruction field, 56d and the data referred to is contained in the data field 52d. Assume the particular instruction in Fig. 10 desires to put to sleep all cells above the identification number

5 6. Therefore the data in data field 52d will be a 6. Once this operation has been carried out by the ALU, the next command word 50e, Fig. 11, may contain in the condition code field 58d the condition, "if greater than put the cells to sleep (if GT sleep)". This capacity to put to sleep cells that are not being used, contributes greatly to the lower power consumption of the SIMD according to this invention. The condition code field 58d actually includes two flags, an AZ flag which

10 indicates ALU=0 and AN which indicates ALU is negative. These flags are used constantly to monitor the status of each of the cells. It can be used in a variety of ways not here pertinent. In this particular case, to induce the sleep mode, if the flags indicate that it is not 0 and not negative with respect to the ALU then it must be greater and if its greater than the ALU (which is 6), according to the operation described in Figs. 10 and 11, the cell will be put to sleep.

15 Although in the example thus far the memories are used to full capacity and they all start loading neatly in the upper left corner, these are not necessary limitations of the invention. Only a portion of each memory need be used and loading could be anywhere. The example so far uses a "two-dimensional" data stream, that is it represents an image but this is not a limitation. The data stream could as well be a one-dimensional data stream such as sound for example with

20 the same applicability and advantages. In the example thus far, too, the data stored in the memories is neatly contiguous but this need not be either. In the case illustrated in Fig. 12, where the memories overlap, the Horiz and Vert lengths are unaffected but Hlen is less than twice the Horiz value as it was before. One benefit of this overlapping is that in the single

overlap areas 100, 102, 104, 106 one piece of data actually loads in two memories. In the double overlap area 108 one piece of data loads in four memories simultaneously which decreases overall loading time.

Although specific features of the invention are shown in some drawings and not in

5 others, this is for convenience only as each feature may be combined with any or all of the other features in accordance with the invention. The words "including", "comprising", "having", and "with" as used herein are to be interpreted broadly and comprehensively and are not limited to any physical interconnection. Moreover, any embodiments disclosed in the subject application are not to be taken as the only possible embodiments.

10 Other embodiments will occur to those skilled in the art and are within the following claims:

What is claimed is: